

```
graph LR; A[INPUT DOCUMENT IMAGE] --> B[LINE ORIENTATION DETERMINING UNIT 1]; A --> C[CHARACTER IMAGE EXTRACTING UNIT 2]; B --> D[CHARACTER ORIENTATION DETECTING UNIT 3]; C --> D; D --> E[DOCUMENT IMAGE CORRECTING UNIT 4]; E --> F[PROPERLY ORIENTATED DOCUMENT IMAGE];
```

The flowchart illustrates the process of document image correction. It begins with an 'INPUT DOCUMENT IMAGE' which is split into two paths. The first path leads to a 'LINE ORIENTATION DETERMINING UNIT' (labeled 1). The output of this unit is fed into a 'CHARACTER ORIENTATION DETECTING UNIT' (labeled 3). The second path from the input image leads to a 'CHARACTER IMAGE EXTRACTING UNIT' (labeled 2), which also feeds into the 'CHARACTER ORIENTATION DETECTING UNIT' (3). The output of the 'CHARACTER ORIENTATION DETECTING UNIT' (3) is then fed into a 'DOCUMENT IMAGE CORRECTING UNIT' (labeled 4). Finally, the output of the 'DOCUMENT IMAGE CORRECTING UNIT' (4) is the 'PROPERLY ORIENTATED DOCUMENT IMAGE'.

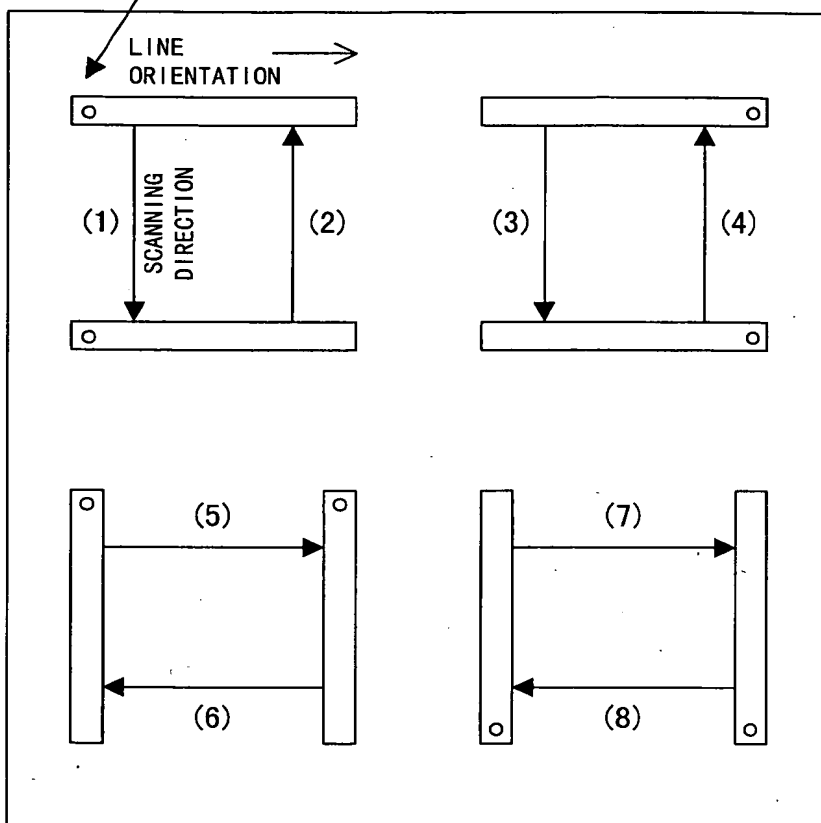
FIG. 1

```
graph LR; Input[INPUT IMAGE] --> 11[IMAGE STORING UNIT]; 11 --> 12[ENGLISH IDENTIFYING UNIT]; 11 --> 13[LINE ORIENTATION DETECTING UNIT]; 11 --> 14[DOCUMENT ORIENTATION DETECTING UNIT]; 11 --> 15[IMAGE CORRECTING UNIT]; 12 --> 14; 13 --> 14; 14 --> 15; 15 --> Output[PROPERLY ORIENTATED IMAGE];
```

The diagram illustrates a document orientation detecting system. It begins with an **INPUT IMAGE** entering an **IMAGE STORING UNIT** (11). From unit 11, the signal is distributed to four parallel processing units: an **ENGLISH IDENTIFYING UNIT** (12), a **LINE ORIENTATION DETECTING UNIT** (13), a **DOCUMENT ORIENTATION DETECTING UNIT** (14), and an **IMAGE CORRECTING UNIT** (15). Both units 12 and 13 provide input to unit 14. The output of unit 14 is then fed into unit 15, which produces the final **PROPERLY ORIENTATED IMAGE**.

FIG. 2

SHEET SCANNER HEAD
OF PAPER /



F I G. 3

(1)



PROPERLY
ORIENTATED

(2)



ROTATED BY
180 DEGREES
AND FLIPPED
TO MIRROR
IMAGE

(3)



FLIPPED TO
MIRROR
IMAGE

(4)



ROTATED BY
180 DEGREES

(5)



ROTATED BY
90 DEGREES
AND FLIPPED
TO MIRROR
IMAGE

(6)



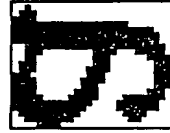
ROTATED BY
270 DEGREES

(7)



ROTATED BY
90 DEGREES

(8)



ROTATED BY
270 DEGREES
AND FLIPPED
TO MIRROR
IMAGE

FIG. 4

PART OF INPUT IMAGE

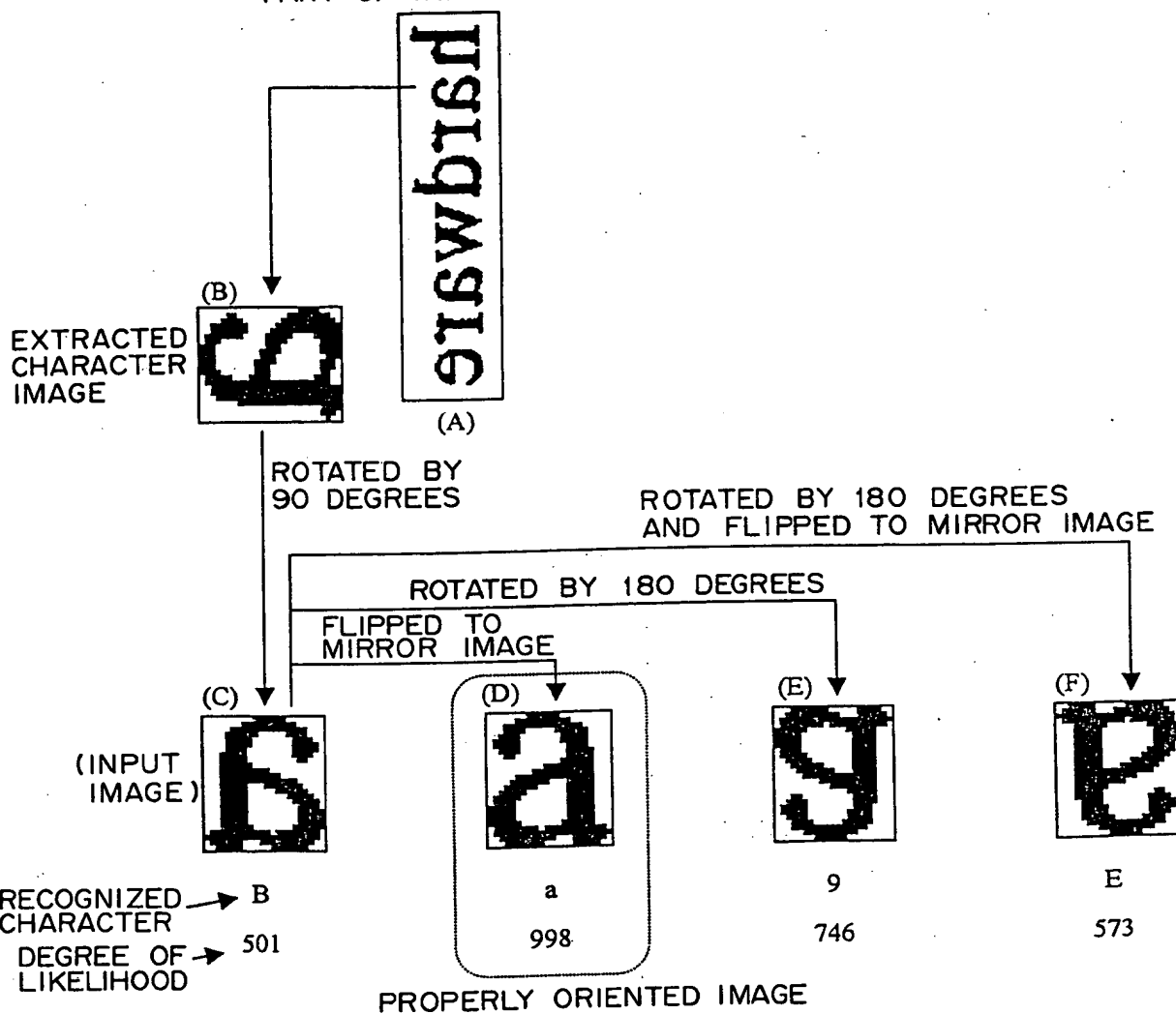


FIG. 5

004070" 092269

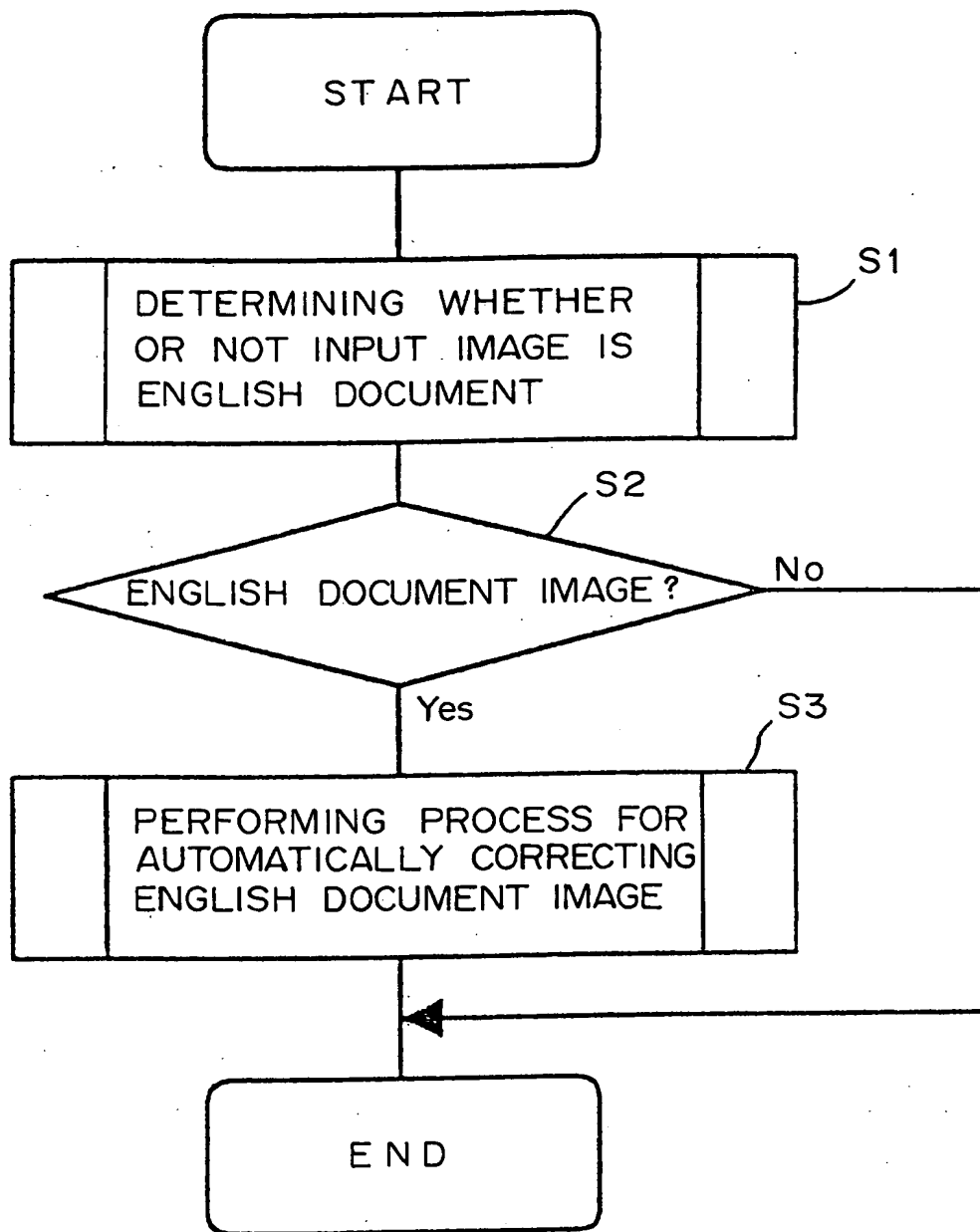


FIG. 7

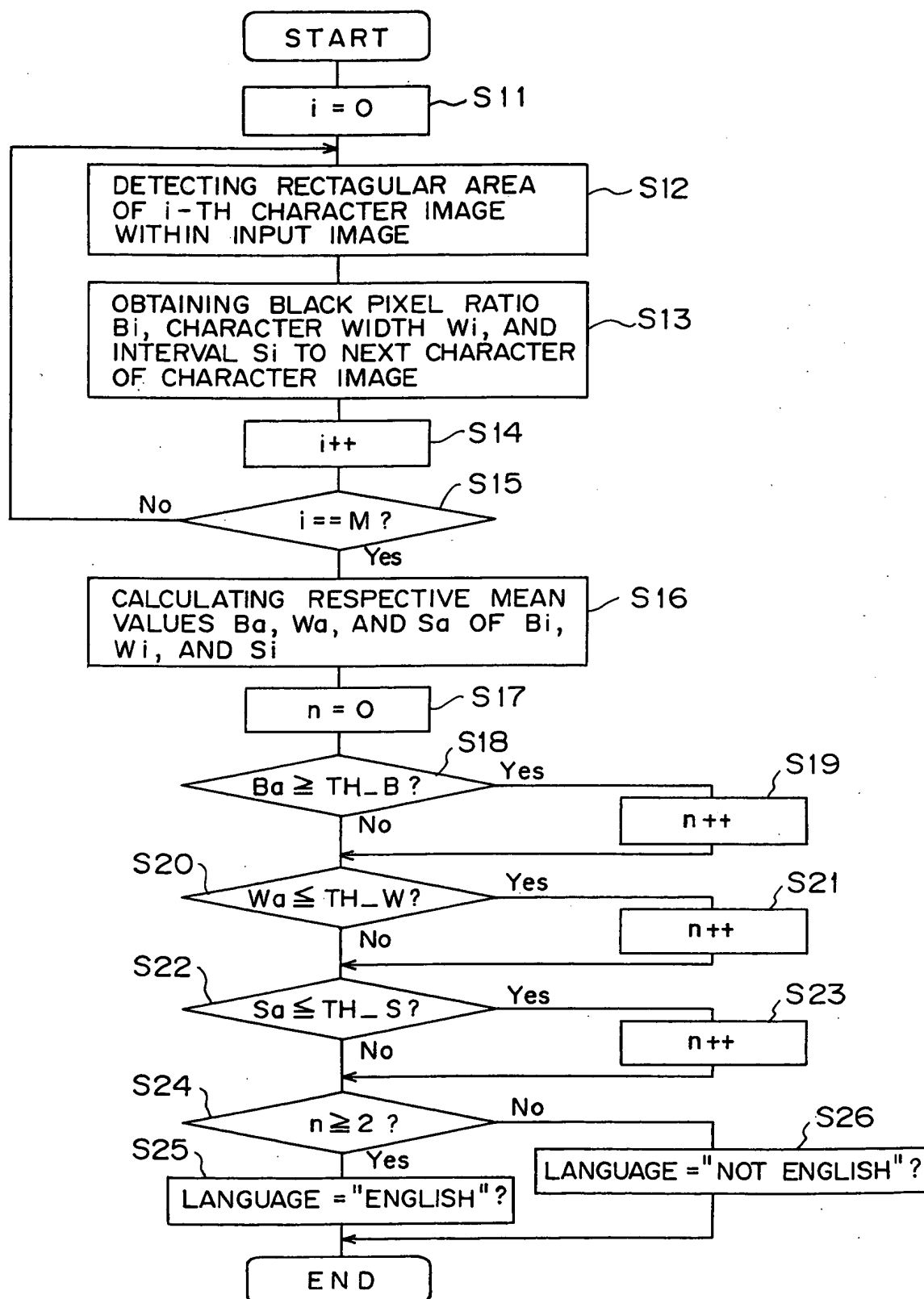


FIG. 8

hardware

CHARACTER WIDTH.

FIG. 9

[illegible]


```

graph TD
    START([START]) --> S31[i=0]
    S31 --> S32[DETECTING RECTANGULAR AREA OF i-TH CHARACTER IMAGE WITHIN INPUT IMAGE]
    S32 --> S33[OBTAINING INTERVAL BETWEEN CHARACTER IMAGES]
    S33 --> S34[i++]
    S34 --> S35{i==M?}
    S35 -- NO --> S32
    S35 -- YES --> S36[CALCULATING DISTRIBUTION "V" OF CHARACTER INTERVALS S1 THROUGH SM]
    S36 --> S37{V ≥ TH_V?  
(DISTRIBUTION IS EQUAL TO OR LARGER THAN THRESHOLD VALUE?)}
    S37 -- YES --> S38[LANGUAGE="ENGLISH"]
    S37 -- NO --> S39[LANGUAGE="NOT ENGLISH"]
    S38 --> END([END])
    S39 --> END
  
```

FIG. 11

```

graph TD
    START([START]) --> S41[S41: DETECTING LINE ORIENTATION OF INPUT IMAGE]
    S41 --> S42[S42: j=0, D1~D4=0]
    S42 --> S43[S43: EXTRACTING ONE CHARACTER IMAGE E1 FROM INPUT IMAGE]
    S43 --> S44{S44: LINE ORIENTATION = VERTICAL?}
    S44 -- Yes --> S45[S45: ROTATING CHARACTER IMAGE E1 BY 90 DEGREES]
    S44 -- No --> S46{S46: RATIO OF LONG SIDE TO SHORT SIDE < 2?}
    S45 --> S46
    S46 -- Yes --> S47{S47: BLACK PIXEL RATIO IS WITHIN RANGE FROM 42 TO 80%?}
    S46 -- No --> S47
    S47 -- No --> S43
    S47 -- Yes --> S48[S48: GENERATING IMAGES E2 THROUGH E4, WHICH ARE OBTAINED BY FLIPPING CHARACTER IMAGE E1 TO MIRROR IMAGE, BY ROTATING E1 BY 180 DEGREES, AND BY ROTATING E1 BY 180 DEGREES AND FLIPPING ROTATED IMAGE TO MIRROR IMAGE]
    S48 --> S49[S49: PERFORMING CHARACTER RECOGNITION FOR E1 THROUGH E4, AND OBTAINING DEGREES OF LIKELIHOOD K1 THROUGH K4 AND CHARACTER CODES C1 THROUGH C4]
    S49 --> S50{S50: TWO OR MORE AMONG K1 THROUGH K4 ARE HIGHER THAN TH_K?}
    S50 -- Yes --> S43
    S50 -- No --> S51[S51: DETECTING MAXIMUM VALUE OF K1 THROUGH K4, AND DEFINING ITS NUMBER TO "M"]
    S51 --> S52{S52: Km ≥ TH_K?}
    S52 -- No --> S43
    S52 -- Yes --> S53{S53: IS Cm ENGLISH CHARACTER?}
    S53 -- No --> S43
    S53 -- Yes --> S54{S54: IS Cm PARTICULAR CHARACTER?}
    S54 -- Yes --> S55[S55: Dm++ ]
    S54 -- No --> S55
    S55 --> S56{S56: Dm == 2?}
    S56 -- No --> S43
    S56 -- Yes --> S59[S59: PERFORMING SAME CONVERSION METHOD AS THAT FOR CONVERTING IMAGE E1 TO IMAGE Em FOR INPUT IMAGE]
    S59 --> END([END])
    S57[S57: j++] --> S58{S58: j == N?}
    S58 -- Yes --> END
    S58 -- No --> S43

```

FIG. 12

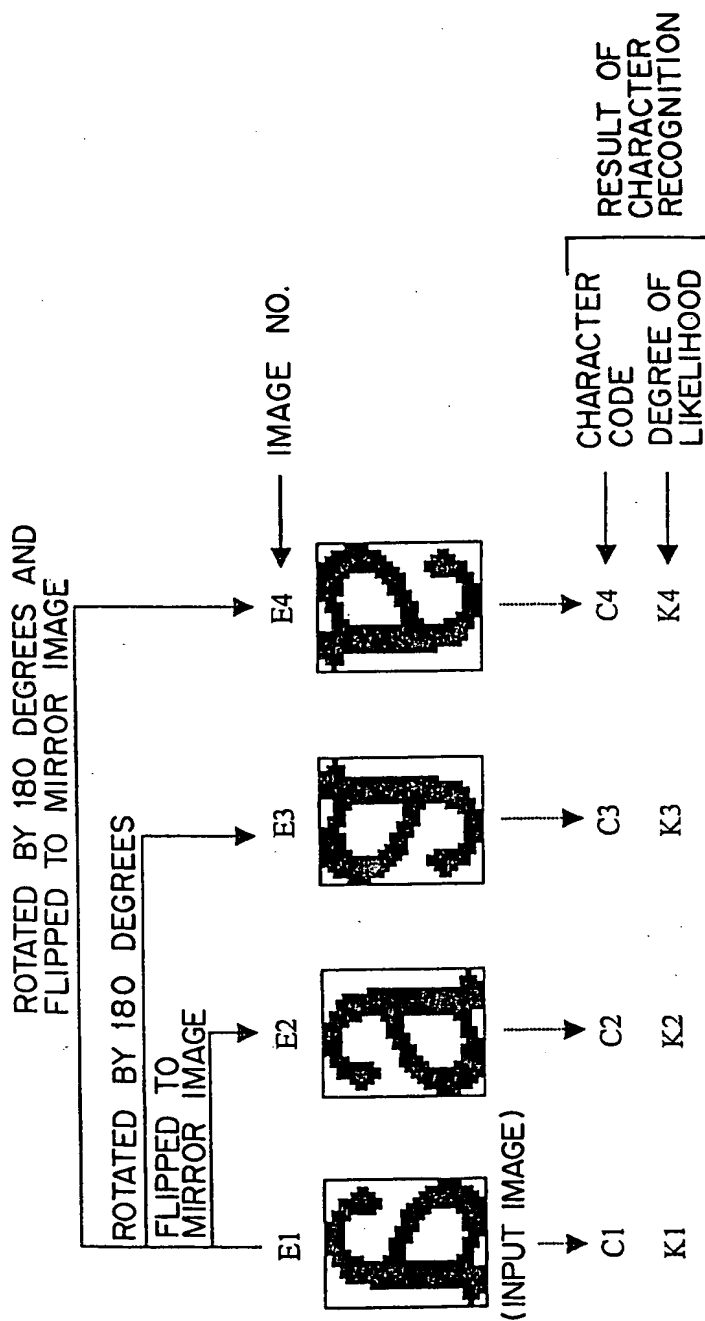


FIG. 13

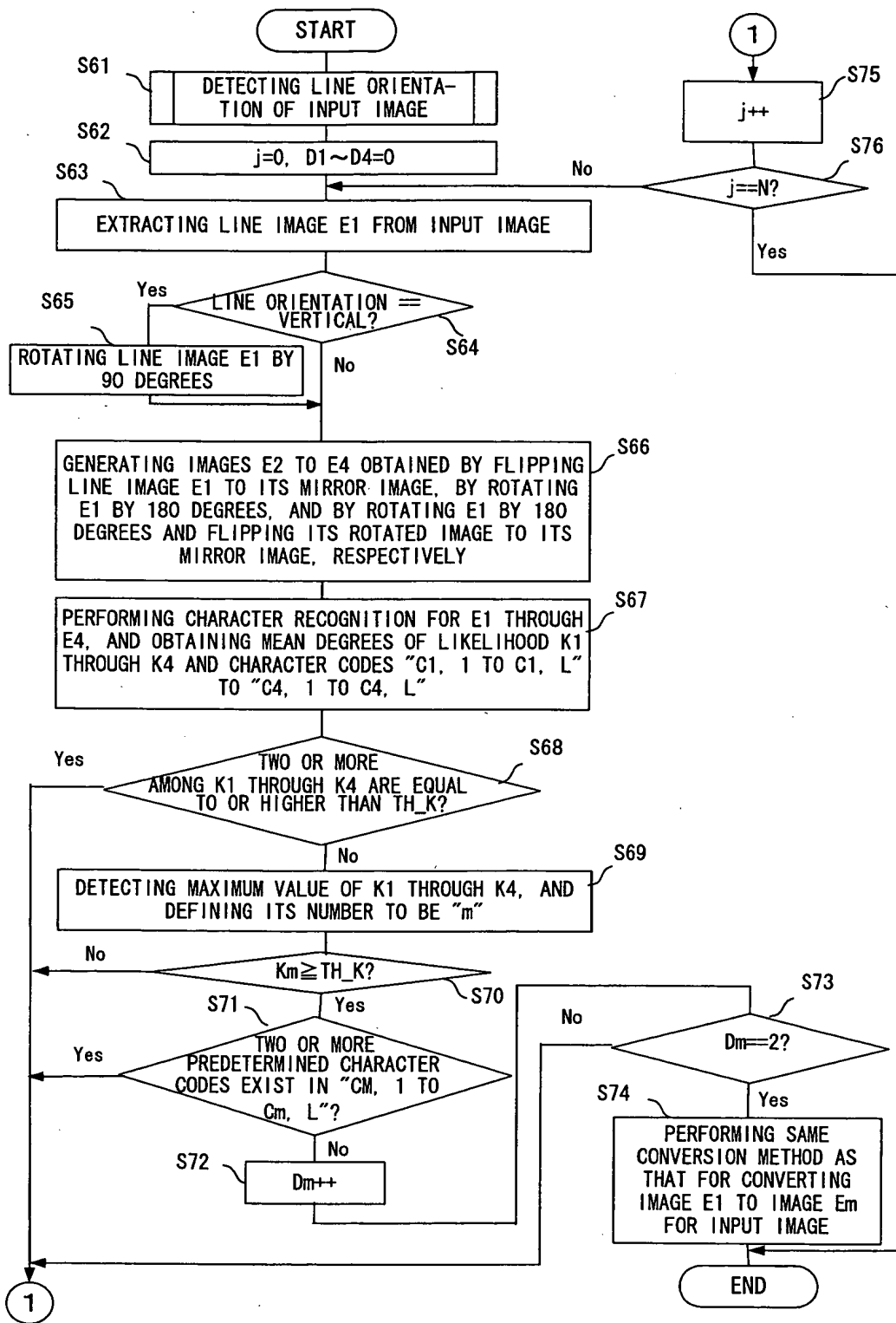


FIG. 14

002020" 2222T950

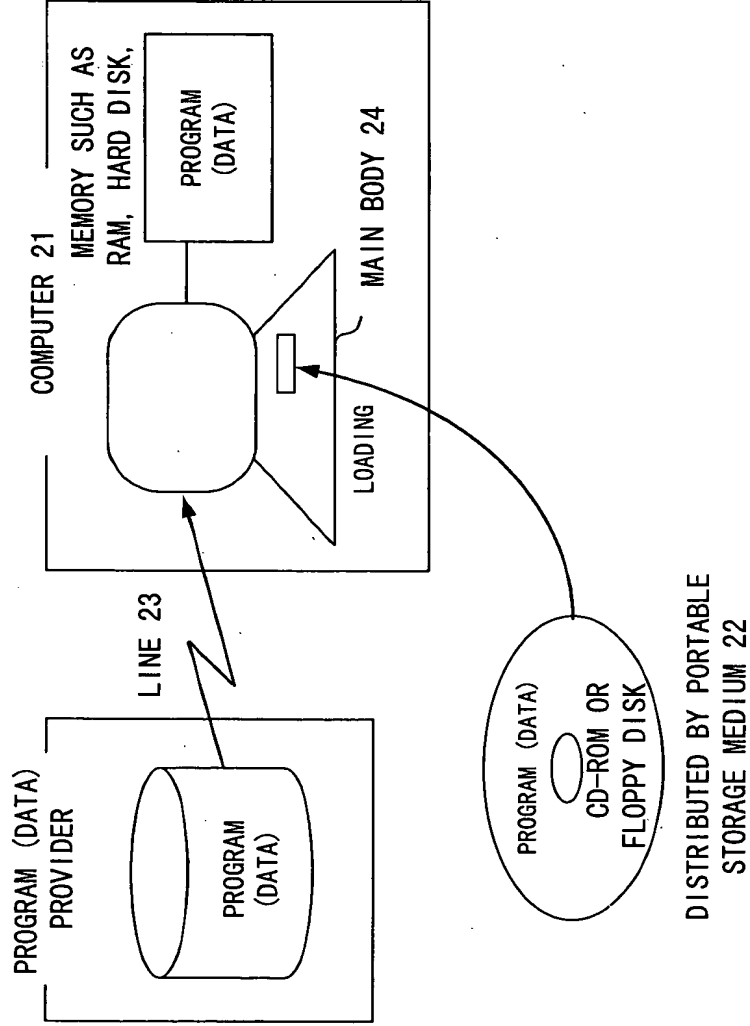


FIG. 15